

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 December 2000 (07.12.2000)

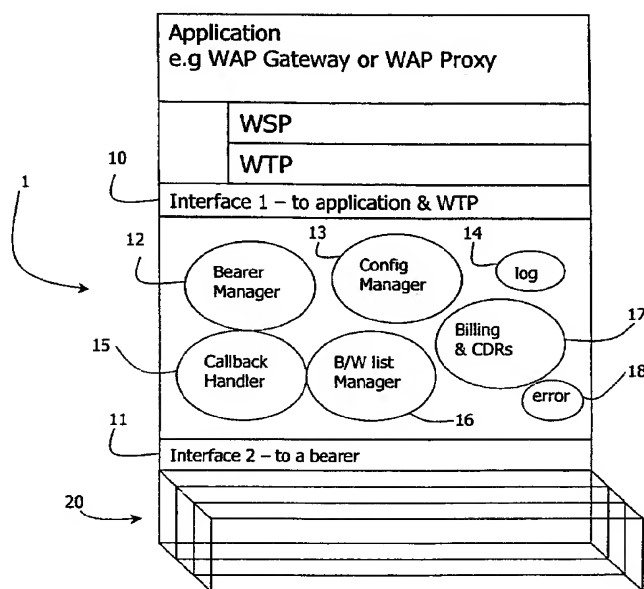
PCT

(10) International Publication Number
WO 00/74342 A1

- (51) International Patent Classification⁷: **H04L 29/06**, H04Q 7/22
- (21) International Application Number: PCT/IE00/00070
- (22) International Filing Date: 1 June 2000 (01.06.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
990451 1 June 1999 (01.06.1999) IE
- (71) Applicant (for all designated States except US): **MARK-PORT LIMITED** [IE/IE]; Custom House Plaza 5, Harbourmaster Place, Dublin 1 (IE).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **SULLIVAN, Owen** [IE/IE]; Pollnaroona, Knocknacarra, Galway (IE). **DOYLE, John** [IE/IE]; 74 Deepdales, Southern Cross Road, Bray, County Wicklow (IE).
- (74) Agents: **WELDON, Michael, J.** et al.; John A. O'Brien & Associates, 3rd floor, Duncairn House, 14 Carysfort Avenue, Blackrock, County Dublin (IE).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**
- With international search report.
 - Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.

[Continued on next page]

(54) Title: BEARER ACCESS IN WIRELESS COMMUNICATION SYSTEMS



(57) Abstract: An interface layer (1) resides on a wireless server hosting server applications. The layer (1) allows bearer modules (20) to be bound to the server and the modules (20) may have a private or a public interface on the bearer system side. The layer (1) has bearer manager (12), configuration manager (13), log (14), callback handler (15), blacklist and whitelist manager (16), billing (17) and error (18) functions.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

"Bearer Access in Wireless Communication Systems"INTRODUCTION5 Field of the Invention

The invention relates to access of servers to Wireless Application Protocol (WAP) bearer systems.

10 At present, it is known to provide an interface, and a development environment for such an interface, which allows access to Short Message Service Centres (SMSCs) via the (Short Message Peer-to-Peer) SMPP protocol. These products are very effective for the particular interfacing involved. However, many developers for environments such as Web server applications are unfamiliar with wireless
15 interfacing functionality. Also, those developers who are expert in this area often prefer to maintain the bearer-side interfacing functionality private and avoid the problems associated with providing backwards and forwards compatibility and ensuring correct use. Such bearer developers also have the problem of testing their interfaces on multiple server platforms.

20

The invention is directed towards providing a bearer interface layer to address these problems.

SUMMARY OF THE INVENTION

25

According to the invention, there is provided an interface layer for a wireless server hosting server applications, the interface layer comprising:

a uniform interface to the wireless server,

30

- 2 -

a bearer module interface for binding bearer modules to provide access of the server applications to bearer systems, and

5 functions between said layers to allow server applications to make calls to connect to a bearer module and to determine status of accessible bearer systems.

In one embodiment, the functions comprises a configuration manager comprising means for maintaining configuration data relating to bearer modules and for
10 responding to server application calls for a configuration list.

In one embodiment, the configuration manager comprises means for returning information regarding network type, bearer system type, and address type to server applications.
15

In one embodiment, the configuration manager comprises means for capturing bearer module information configured by a bearer developer within constraints of interfacing to the bearer module interface.

20 In one embodiment, the configuration manager comprises means for maintaining a configuration file for every bound bearer module.

In one embodiment, the functions include a bearer manager comprising means for registering bearer modules and for providing bearer module status information to
25 requesting server applications.

In one embodiment, the bearer manager comprises means for responding to server application calls to register a bearer module.

- 3 -

In one embodiment, the bearer manager comprises means for making a bind attempt to bind a bearer module to a transport mechanism.

5 In one embodiment, the bearer manager comprises means for responding to application calls to disconnect from a bearer module.

In one embodiment, the functions include a blacklist and whitelist manager comprising means for automatically testing all non-secure traffic.

10 In one embodiment, the blacklist and whitelist manager comprises means for filtering traffic according to parameters set by a server application.

15 In one embodiment, the functions include a callback handler comprising means for allowing a server application to call routines for pre-processing and post-processing of WDP packets.

20 In one embodiment, the functions include a log function comprising means for logging test data from a bearer module under test, and for providing said data to a requesting server application.

In one embodiment, the functions include a billing function comprising means for generating billing data on the basis of criteria supplied by a server application.

25 According to another aspect, the invention provides a wireless application server comprising WSP and WTP layers over an interface layer as defined above.

DETAILED DESCRIPTION OF THE INVENTION

Brief Description of the Drawings

30

The invention will be more clearly understood from the following description of some embodiments thereof, given by way of example only with reference to the accompanying drawings in which:

5 Fig. 1 is an overview diagram showing the context of an interface layer of the invention;

 Fig. 2 is a diagram showing the structure of the interface layer; and

10 Figs. 3 and 4 are diagrams illustrating uses of the interface layer; and

 Fig. 5 is a diagram illustrating signal interactions.

Description of the Embodiments

15

Referring to Fig. 1, interface layers 1 allow access by wireless servers 2 to bearer systems 3. In the embodiment illustrated, the servers 2 include:-

- a WAP server 5 hosting prototype application and content,
- 20 - a WAP proxy 6 performing protocol conversion and content encoding, and
- a WTA server 7 hosting telephony based applications.

The servers are connected to a Web server 8 via a Firewall 9. The bearer systems are
25 in this embodiment:-

- a Circuit Switched Data bearer system 10,
- an Express Messaging SMSC bearer system 11,
- an SMSC bearer system 12, and
- 30 - a USSD gateway bearer system 13.

Referring to Fig. 2 each interface layer 1 comprises a server interface 10 to the applications and WTP (Wireless Transaction Protocol) layer of the server. The interface 10 is uniform in its interfacing operations to all of the server applications. It
5 also comprises an interface 11 to bearer system interfacing modules 20, called "bearers" or "bearer modules" in this specification. Between these two interfaces there is:

- a bearer manager 12,
- 10 a configuration manager 13,
- a log 14,
- a callback handler 15,
- a blacklist/whitelist manager 16,
- a billing and CDR function 17, and
- 15 an error function 18.

The interface 10 is a uniform interface to all of the server applications and the interface 11 allows bearers 20 to be loaded and offers their services to the applications.

20

Referring to Fig. 3, one of the bearers 20 is a test bearer which writes to the log 14 and the log 14 returns responses to an application 25 on a WAP stack 26. This is a convenient framework for creation of new bearer modules 20. It also serves to examine and explain the facilities offered by an existing bearer 20. It is thus very
25 suitable for testing bearer modules 20 and allows short development times.

Referring to Fig. 4, a WAP gateway 40 having a WAP stack 41 uses the interface layer 1 to access a data gateway 42 and an SMSC 43. Each of these bearer systems is accessed via a bearer module 20 (Bearer A and Bearer B). However the gateway 42
30 is also accessed by a private interface 44 to be kept secret, while the SMSC is

accessed via SMPP 45 (which is public). It will thus be appreciated that the layer 1 allows versatility in the manner in which bearer developers may link their bearer system with a server: a private or a public protocol may be used. All of the bearer modules 20 are bound by the rules of the interface 11, but may or may not be private
5 on the bearer system side. This is illustrated in Fig. 4 in which there is a secret interface 46 and an SMPP interface 47 on the bearer system side. The following describes the functions between the interfaces 10 and 11 in more detail.

Configuration manager 13

10 The manager 13 maintains a file which specifies which bearer modules are present, and there is a configuration file of bearer module parameters for each bearer module.

Bearer modules 20

Each bearer module consists of one or more bearer files and bearer configuration
15 files. Where possible, bearer modules are designed for dynamic loading but where this is not possible or not desirable static linking of bearer modules is supported.

Bearer manager 12

The bearer manager 12 is responsible for the loading and management of the bearer
20 modules. It performs version checking and license checking

Billing & CDRs 17

This can be set to be in automatic operation to generate billing information (and pre-paid queries) for all non secure traffic or it can be invoked by the server application
25 for all types of traffic. Billing is on a basis set by the server application, such as per-URL or per data volume.

White & Blacklist Manager 16

This can be set to be in 'automatic operation' and automatically test all non-secure traffic, or can be invoked by a server application for required types of traffic. The contents of the lists are URLs or originator addresses.

5

Callback Handler 15

This allows the server application to request the layer 1 call its routines to provide extended pre and post processing of WDP packets.

10 The calls available to the server applications are set out in the table below.

wdc_bearer_list	Returns the list of bearer modules 20 configured on the server	
wdc_info	Get information about a specific bearer module.20	
wdc_connect	Connect to a bearer module 20	
wdc_send	Send a message to the specified bearer system	
wdc_get	Get a message from a specified bearer system	
wdc_param	Get details of a configuration parameter from a configuration file	
wdc_log	Create a log entry in the log/trace files.	
wdc_close	Disconnect from a bearer module.	

The following handlers are provided to the bearer developer

wdc_register	Handle the registration of a bearer module.	
wdc_connect	Connect to a bearer module.	
wdc_send	Send a message to the specified bearer	
wdc_get	Get a message from a specified bearer	
wdc_param	Get details of a configuration parameter from a configuration files	
wdc_info	Get information about a specify bearer module.	
wdc_log	Create a log entry in the log/trace files.	
wdc_close	Disconnect from a bearer module.	

- 5 Some of the above calls are illustrated in Fig. 5, in which server applications communicate with an SMSC bearer system for bi-directional transfer of WAP content. The following describes the calls in more detail, with examples of code.

wdc_bearer_list

- 10 Returns the list of bearer modules configured by the system. The call actually returns the value of the bearer list parameter from the wdc-dtk.ini (layer 1 configuration file).

Each item in the bearer list is known as a bearer tag and can be used to retrieve information on that bearer using the “wdc_info” call.

wdc-dtk.ini

5 [main]

Bearer_List = SMSC01, SMSC_DG, USSD, CSD23, EXP-001

Each of the items in the list will have their own section in the “.ini” file.

[SMSC01]

10 bearer_module = “smc01/smc3-4”

parameter_file = “smc01/smc.ini”

network = “GSM”

bearer_type = “SMS”

address_type = “GSM_MSISDN”

15 assigned_number = “0x03”

wdc_info

Returns an INFO element for a connection. The information returned includes the
20 information from the “Network Bearer table” defined by WAP.

- Network
- Bearer type
- Address Type
- Assigned Number

25

A connection is not required to obtain information about the bearer.

When a new bearer module 20 is being created a handler for each of the above calls is provided. How the call is implemented is decided by the developer of each bearer
30 module within the constraints imposed by the interface 11. For example a developer

can chose to implement the “wdc_parm” handler to read from a flat file, to read from a database or to permanently return a blank value.

wdc_register

5

This is called by the application to load/register a specific bearer module. One of the parameters passed in the version identifier of the interface to ensure that only modules of a correct or later version are loaded. If the layer 1 locates an older version of the bearer module then that which it expects an error is returned.

10

wdc_connect

This function is used by the application to connect to a bearer. It makes an application layer connection to the bearer. The bearer module should, on a ‘bind’, attempt to connect/bind to the transport mechanism to ensure it is in place. If an application wants to connect to multiple bearer systems, say USSD and SMS, then it must make multiple bind calls:

15

```
socket = wdc_connect(bearer);
```

20

e.g.

```
socketSMS = wdc_connect(GSM_SMS);
```

Returns

25

Success: This function returns a non-NULL connection identified which will be used by all further call to this bearer until the connection is terminated.

Failure: The function returns null and stores an error in the global error number variable.

30

wdc_close

This tells the bearer module that the server application is disconnecting. All resources allocated to using the bearer should be freed.

5 `wdc_close(socketSMS);`

wdc_send

Is used to send information to a data gateway via a bearer module using the layer 1.

10 **wdc_get**

Is used to read information from a data gateway via a bearer module using the layer 1.

wdc_log

15

`wdc_log(socket, severity, msg);`

An example is:

`wdc_log(socketSMS, WARNING, "Less the 10 % Memory remaining.");`

20 Regarding billing and prepaid queries, one of the areas the layer 1 simplifies is the complex area of interfacing to telecom billing and prepaid systems. It is possible for an application to instruct the layer 1 to load and enable a billing sub module. The application may chose to enable the layer 1 to automatically log location URL and/or Volume of data as CRD for billing. This is only possible for WAP if the non-secure port is used.

25

Wdc_cdr_mode(setting)

- VOLUME
- URL
- 30 - OFF

The INI setting “billing_module” points to the location of the billing module to load.

The application may call the layer 1 to create CDRs.

5

```
wdc_cdr_volume(CDR_HEADER, 400);
```

Log 400 bytes

```
wdc_cdr_url(CDR_HEADER, “http://www.wapforum.com/index.wml”);
```

10

Other calls include Whitelist and Blacklist management, in which an application may enable automatic Black/White list checking. The application may call the layer 1 to check a URL against the Black/While list using :

15

```
wdc_check_mode(ON/OFF);
```

```
wdc_check_url(check_interface, url);
```

The INI parameter:

20

“Black_while_list” indicates the file to use.

wdc_parm

25

This is used by applications for parameter reading:

```
err_code_t wdc_parm(bearer_tag, wdc_param_t, *parameter);
```

```
typedef struct
```

```
{
```

30

```
    char name[WDC_PARAM_NAME_SIZE];
```

```
    char value[WDC_PARAM_VALUE_SIZE];
```

```
} wdc_pararm_t;
```

- This call is used to get a value of a configuration parameter from one of the configuration files. The wdc_module parameter selects which of the module –
- 5 MAIN or one of the bearers is queried. Having a separate configuration file for each bearer module is necessary to prevent naming conflicts when bearer modules are provided by third parties.

RETURNS

- 10 Success: The function returns and ERR_OK and the value of the parameter is placed in the value field of the structure.

Failure: The function returns a non-zero error number.

- 15 Callbacks are provided by the layer 1 to allow an application request that it handles various events:

```
wdc_callback(item, handler)
```

- 20 ERROR – allows the application register its own handler for error conditions this is useful if the application already has an error logging facility that it wishes to use.

LOG – allow an application to register its own handler for logging messages.

- 25 It will be appreciated that the invention provides for excellent versatility for both bearer-side and server-side development. The interface layer 1 also allows the addition of new bearers as they become available, and the application to utilise ‘intelligent’ selection of bearers.

- 30 The invention is not limited to the embodiments described, but may be varied in construction and detail.

Claims

1. An interface layer for a wireless server hosting server applications, the interface layer comprising:
5 a uniform interface (10) to the wireless server,

a bearer module interface(11) for binding bearer modules (20) to provide access of the server applications to bearer systems, and
10 functions between said layers to allow server applications to make calls to connect to a bearer module and to determine status of accessible bearer systems.
- 15 2. An interface layer as claimed in claim 1, wherein the functions comprises a configuration manager comprising mean for maintaining a configuration file of bearer modules and for responding to server application calls for a configuration list.
- 20 3. An interface layer as claimed in claim 2, wherein the configuration manager comprises means for returning information regarding network type, bearer system type, and address type to server applications.
- 25 4. An interface layer as claimed in claim 3, wherein the configuration manager comprises means for capturing bearer module information configured by a bearer developer within constraints of interfacing to the bearer module interface.

- 15 -

5. An interface layer as claimed in claims 2 to 4, wherein the configuration manager comprises means for maintaining a configuration file for every bound bearer module.
- 5 6. An interface layer as claimed in any preceding claim wherein the functions include a bearer manager comprising means for registering bearer modules and for indicating bearer module status.
7. An interface layer as claimed in claim 6, wherein the bearer manager comprises
10 means for responding to server application calls to register a bearer module.
8. An interface layer as claimed in claim 7, wherein the bearer manager comprises means for making a bind attempt to bind a bearer module to a transport mechanism.
15
9. An interface layer as claimed in claims 6, 7, or 8 wherein the bearer manager comprises means for responding to application calls to disconnect from a bearer module.
- 20 10. An interface layer as claimed in any preceding claim, wherein the functions include a blacklist and whitelist manager comprising means for automatically testing all non-secure traffic.
11. An interface layer as claimed in claim 10, wherein the blacklist and whitelist
25 manager comprises means for filtering traffic according to parameters set by a server application.
12. An interface layer as claimed in any preceding claim wherein the functions include a callback handler comprising means for allowing a server application
30 to call routines for pre-processing and post-processing of WDP packets.

13. An interface layer as claimed in any preceding claim, wherein the functions include a log function comprising means for logging test data from a bearer module under test, and for providing said data to a requesting server application.
- 5
14. An interface layer as claimed in any preceding claim, wherein the functions include a billing function comprising means for generating billing data on the basis of criteria supplied by a server application.
- 10
15. A wireless application server comprising a WSP and WTP layers over an interface layer as claimed in any preceding claim.

1/5

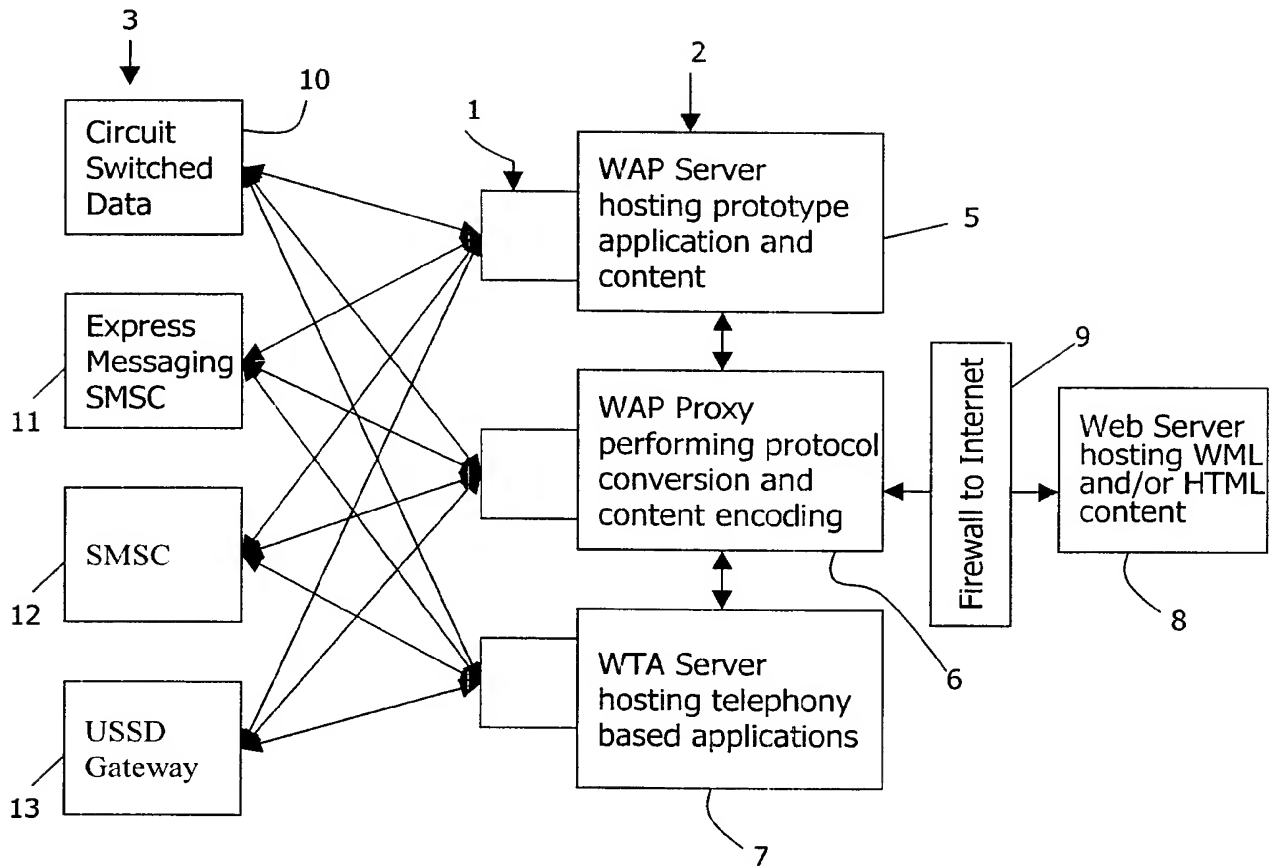


Fig. 1

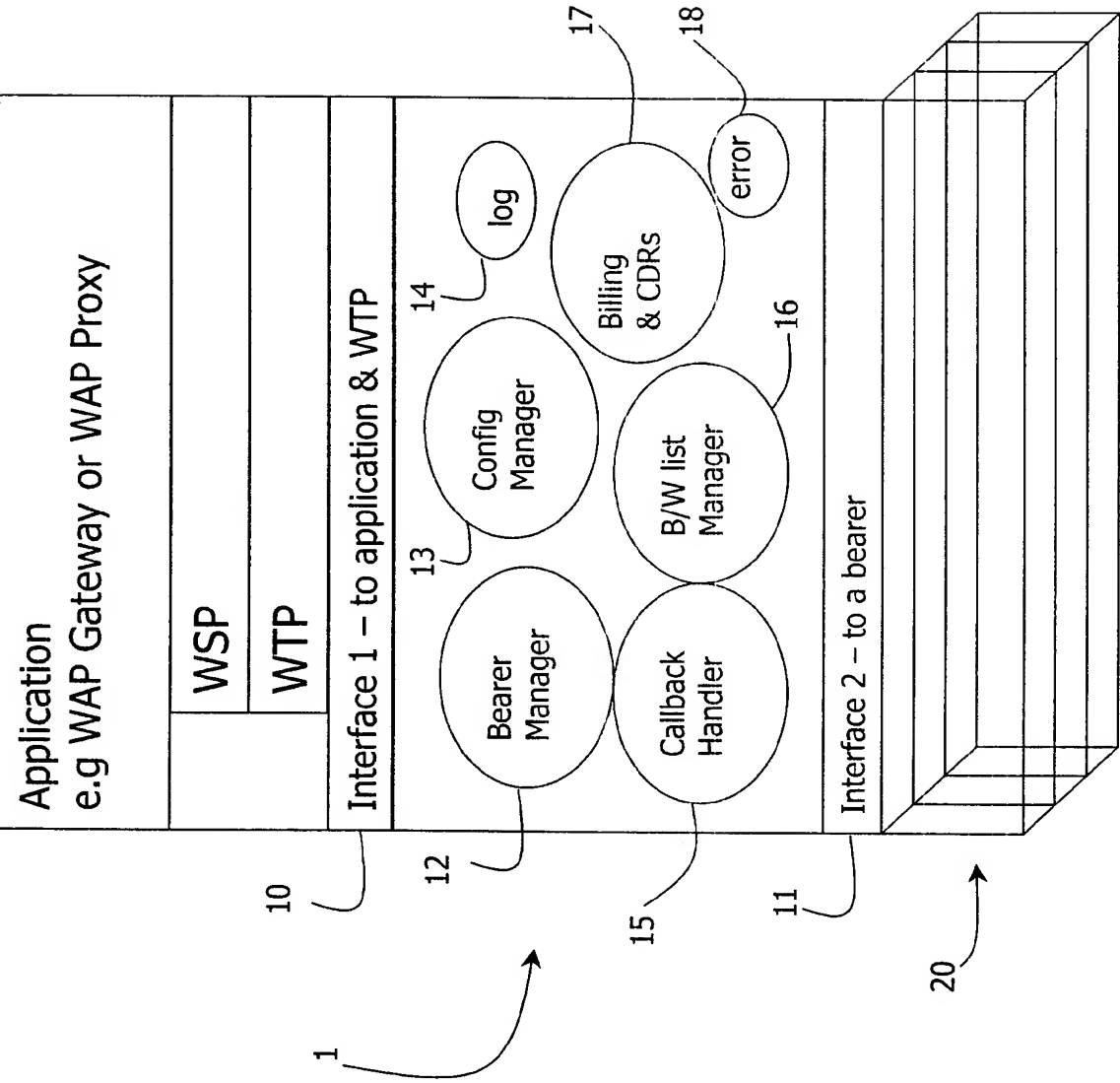


Fig. 2

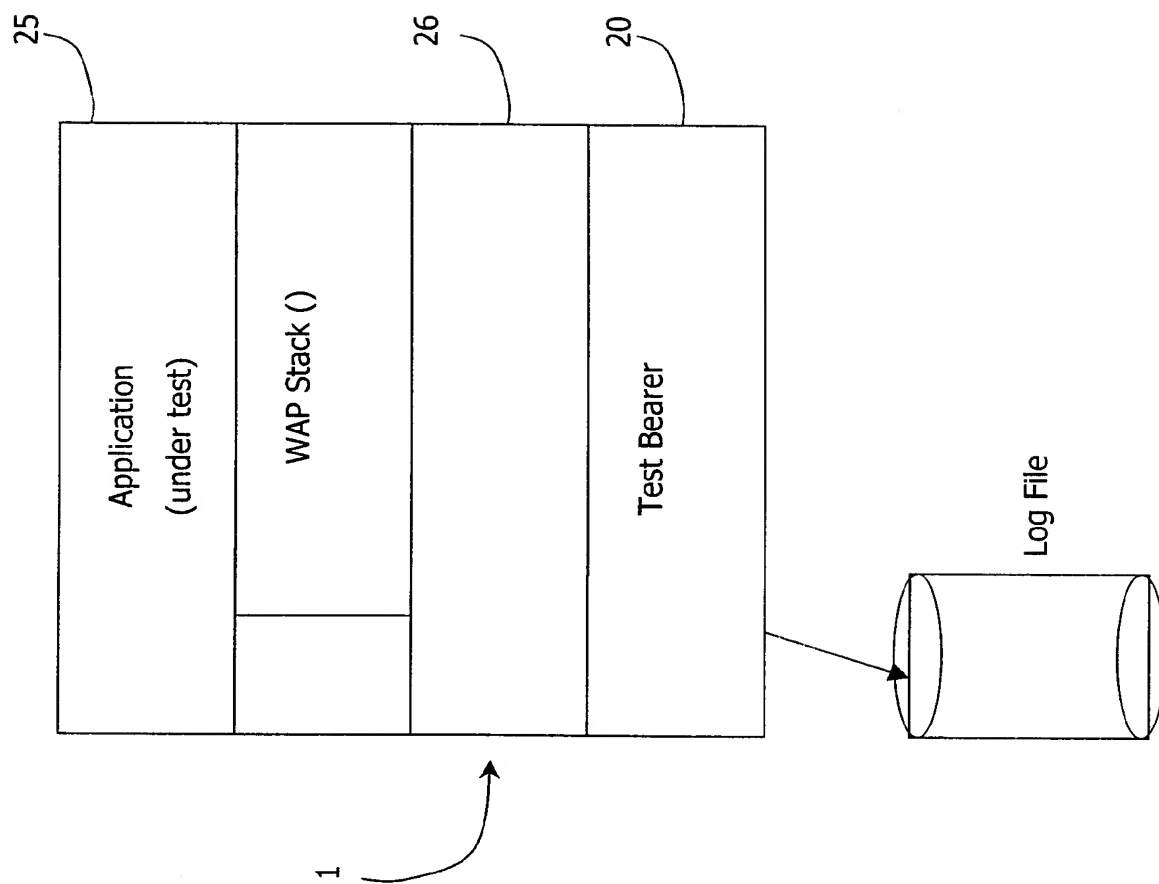


Fig. 3

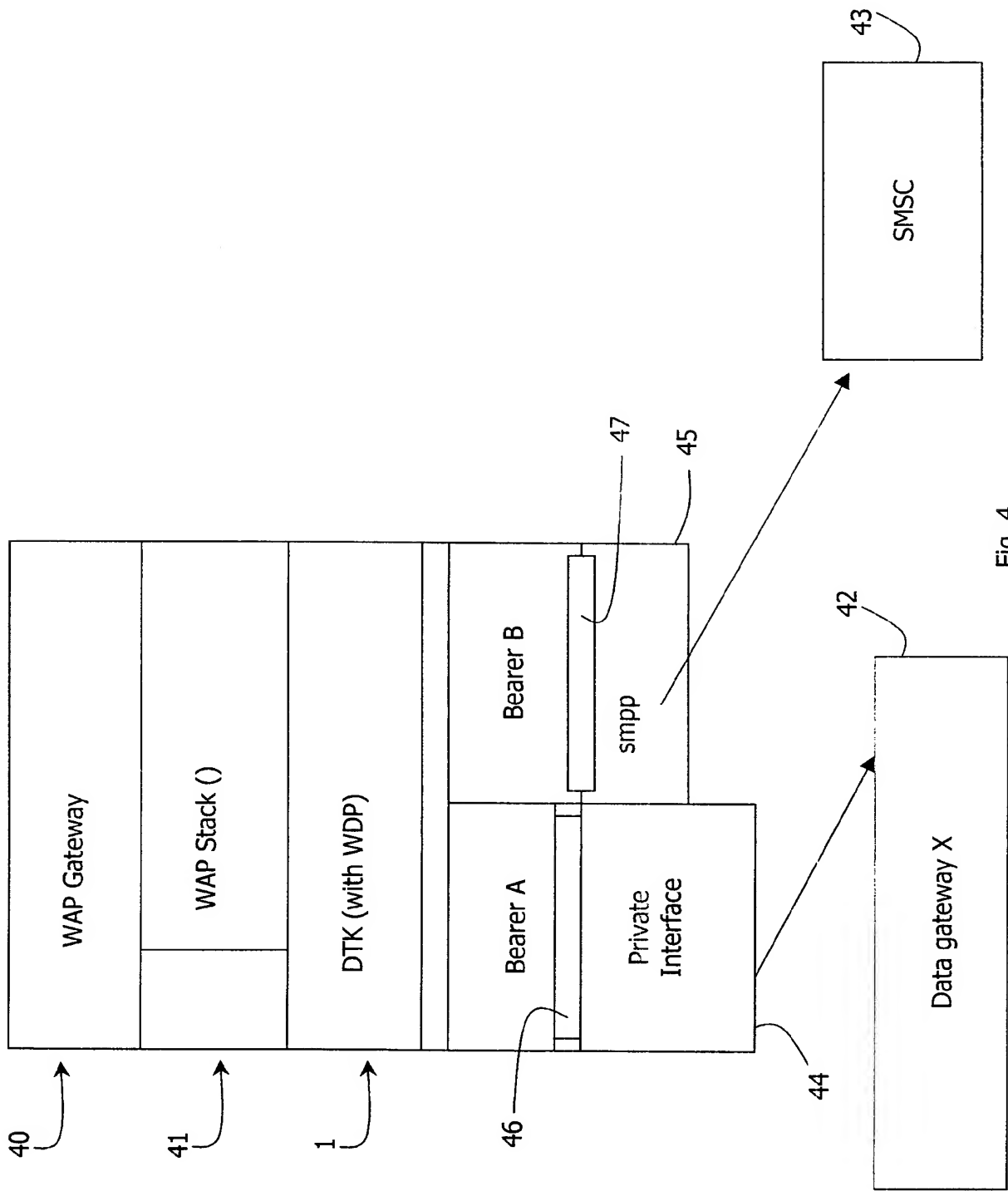


Fig. 4

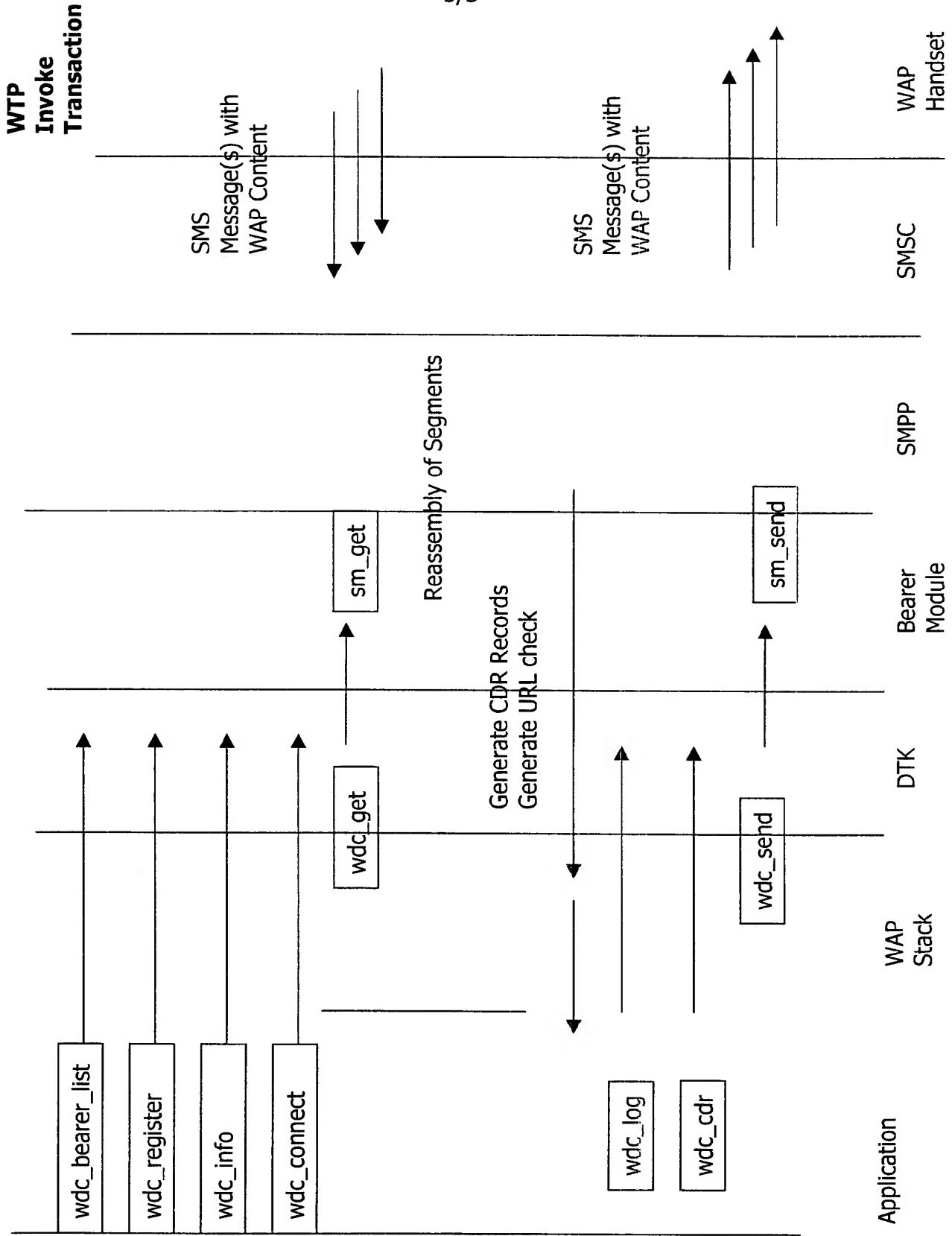


Fig. 5

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IE 00/00070

A. CLASSIFICATION OF SUBJECT MATTER IPC 7 H04L29/06 H04Q7/22		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 H04L H04Q		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data, PAJ, INSPEC		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A A	WO 99 14877 A (MOTOROLA INC) 25 March 1999 (1999-03-25) page 1, paragraph 4 -page 2, paragraph 4 page 4, line 4 -page 8, line 13 figures 1,2 --- MARTINI G ET AL: "DISTRIBUTED ARCHITECTURE FOR APPLICATIONS BASED ON THE GSM SHORT MESSAGE SERVICE" INTERNATIONAL WORKSHOP ON SERVICES IN DISTRIBUTED AND NETWORKED ENVIRONMENTS,XX,XX, 5 June 1995 (1995-06-05), pages 140-145, XP000612264 page 143, right-hand column, line 1 - line 20 --- <div style="text-align: center;">-/--</div>	1-3,6-9, 15 4,5, 10-14 12-14
<div style="display: flex; justify-content: space-between;"> <input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex. </div>		
* Special categories of cited documents : <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </div> <div style="width: 45%;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&" document member of the same patent family</p> </div> </div>		
Date of the actual completion of the international search <div style="text-align: center; font-weight: bold;">2 October 2000</div>		Date of mailing of the international search report <div style="text-align: center; font-weight: bold;">09/10/2000</div>
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer <div style="text-align: center; font-weight: bold;">Blanco Cardona, P</div>

INTERNATIONAL SEARCH REPORT

International Application No

PCT/IE 00/00070

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>TURO HEIKKINEN: "Wireless Application Protocol" MULTIMEDIASEMINAARI, 8 - 8 April 1999, pages 1-7, XP002148057 page 3, paragraph 5 -page 4, paragraph 7 -----</p>	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/IE 00/00070

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
W0 9914877 A	25-03-1999	AU 8685598 A FR 2769455 A	05-04-1999 09-04-1999
